REFERENCES: https://www.kaggle.com/shivamsarawagi/wildimagedetection-0-875 (https://www.kaggle.com/shivamsarawagi/wildimagedetection-0-875) https://www.kaggle.com/gowrishankarin/eda-with-plotly-smart-cute-and-pretty-people (https://www.kaggle.com/gowrishankarin/eda-with-plotly-smart-cute-and-pretty-people) https://www.kaggle.com/vaishvik25/blend-of-smiles (https://www.kaggle.com/vaishvik25/blend-of-smiles)

Motivation:

AAIC team

Note:

As AAIC always do I am doing everything in train data\ This run on kaggle kernal, but bargraph was not visible in colab so I took screenshot and merged.

Private leader rank 55

after competition end under 20

Reading the directory

In [0]:
```
# This Python 3 environment comes with many helpful analytics libraries
 installed
# It is defined by the kaggle/python docker image: https://github.com/ka
ggle/docker-python
# For example, here's several helpful packages to load in

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the "../input/" directory.
# For example, running this (by clicking run or pressing Shift+Enter) wi
ll list the files in the input directory

import os
print(os.listdir("../input"))

# Any results you write to the current directory are saved as output.
```

```
['test', 'train_relationships.csv', 'train', 'sample_submission.csv']
```

Extracting pretrained vggFaces [https://github.com/rcmalli/keras-vggface (https://github.com/rcmalli/keras-vggface)](https://github.com/rcmalli/keras-vggface)

In [0]:
```
!pip install git+https://github.com/rcmalli/keras-vggface.git
```

```
Collecting git+https://github.com/rcmalli/keras-vggface.git
  Cloning https://github.com/rcmalli/keras-vggface.git to /tmp/pip-req-
build-jah0ms4_
  Running command git clone -q https://github.com/rcmalli/keras-vggfac
e.git /tmp/pip-req-build-jah0ms4_
Requirement already satisfied (use --upgrade to upgrade): keras-vggface
==0.6 from git+https://github.com/rcmalli/keras-vggface.git in /opt/con
da/lib/python3.6/site-packages
Requirement already satisfied: numpy>=1.9.1 in /opt/conda/lib/python3.
6/site-packages (from keras-vggface==0.6) (1.16.4)
Requirement already satisfied: scipy>=0.14 in /opt/conda/lib/python3.6/
site-packages (from keras-vggface==0.6) (1.2.1)
Requirement already satisfied: h5py in /opt/conda/lib/python3.6/site-pa
ckages (from keras-vggface==0.6) (2.9.0)
Requirement already satisfied: pillow in /opt/conda/lib/python3.6/site-
packages (from keras-vggface==0.6) (5.4.1)
Requirement already satisfied: keras in /opt/conda/lib/python3.6/site-p
ackages (from keras-vggface==0.6) (2.2.4)
Requirement already satisfied: six>=1.9.0 in /opt/conda/lib/python3.6/s
ite-packages (from keras-vggface==0.6) (1.12.0)
Requirement already satisfied: pyyaml in /opt/conda/lib/python3.6/site-
packages (from keras-vggface==0.6) (5.1.1)
Requirement already satisfied: keras-applications>=1.0.6 in /opt/conda/
lib/python3.6/site-packages (from keras->keras-vggface==0.6) (1.0.8)
Requirement already satisfied: keras-preprocessing>=1.0.5 in /opt/cond
a/lib/python3.6/site-packages (from keras->keras-vggface==0.6) (1.1.0)
Building wheels for collected packages: keras-vggface
  Building wheel for keras-vggface (setup.py) ... done
  Created wheel for keras-vggface: filename=keras_vggface-0.6-cp36-none
-any.whl size=8311 sha256=d9bb48bd0e209f8070aa9d5a6b8cd8da7d2c909f1a669
050ba332241dcfb21b8
  Stored in directory: /tmp/pip-ephem-wheel-cache-0_1ygax3/wheels/36/0
7/46/06c25ce8e9cd396dabe151ea1d8a2bc28dafcb11321c1f3a6d
Successfully built keras-vggface
```

Importing all the necessary libraries

In [0]:
```
from collections import defaultdict
from glob import glob
from random import choice, sample

import cv2
import numpy as np
import pandas as pd
from keras.callbacks import ModelCheckpoint, ReduceLROnPlateau
from keras.layers import Input, Dense, Flatten, GlobalMaxPool2D, GlobalA
vgPool2D, Concatenate, Multiply, Dropout, Subtract, Add, Conv2D
from keras.models import Model
from keras.preprocessing import image
from keras.optimizers import Adam
from keras_vggface.utils import preprocess_input
from keras_vggface.vggface import VGGFace
import h5py
```

https://www.kaggle.com/gowrishankarin/eda-with-plotly-smart-cute-and-pretty-people (https://www.kaggle.com/gowrishankarin/eda-with-plotly-smart-cute-and-pretty-people) **HOld tight we are going to see some basic but important visualization**

> *https://plot.ly/python/ (https://plot.ly/python/)*

- *Plotly's Python graphing library makes interactive, publication-quality graphs. Examples of how to make line plots, scatter plots, area charts, bar charts, error bars, box plots, histograms, heatmaps, subplots, multiple-axes, polar charts, and bubble charts.*

In [0]:
```
import matplotlib.pyplot as plt
from PIL import Image

import plotly.offline as py
py.init_notebook_mode(connected=True)
import plotly.graph_objs as go
import plotly.tools as tls

pd.options.mode.chained_assignment = None
pd.options.display.max_columns = 9999
pd.options.display.float_format = '{:20, .2f}'.format
```

LOADING the file from directory

In [0]:
```
train_df = pd.read_csv("../input/train_relationships.csv")
train_df.head()
```

Out[0]:

	p1	p2
0	F0002/MID1	F0002/MID3
1	F0002/MID2	F0002/MID3
2	F0005/MID1	F0005/MID2
3	F0005/MID3	F0005/MID2
4	F0009/MID1	F0009/MID4

In [0]:
```
train_df.shape
```

Out[0]:
```
(3598, 2)
```

In [0]:
```
files = [os.path.join(dp, f) for dp, dn, fn in os.walk(os.path.expanduse
r("../input/train")) for f in fn]
files[0]
```

Out[0]:
```
'../input/train/F0897/MID2/P09464_face1.jpg'
```

Creating a dataframe for further working

Below dataframe contains the the following column:

- files-> It contains all the train images from where we can directly load the train images\
- familyID->It contains the familyID(3rd place from right in files)
- kinID-> It contains the kinID(4th place from right in files)
- iniqueID->It contains uniqueID(on concanating familyID with kinID)

In [0]:
```
train_images = pd.DataFrame({
    'files': files,
    'familyId': [file.split('/')[3] for file in files],
    'kinId': [file.split('/')[4] for file in files],
    'uniqueId': [file.split('/')[3] + '/' + file.split('/')[4] for file
in files]
})
train_images.head()
```

Out[0]:

	files	familyId	kinId	uniqueId
0	../input/train/F0897/MID2/P09464_face1.jpg	F0897	MID2	F0897/MID2
1	../input/train/F0897/MID5/P09469_face1.jpg	F0897	MID5	F0897/MID5
2	../input/train/F0897/MID3/P09464_face2.jpg	F0897	MID3	F0897/MID3
3	../input/train/F0897/MID4/P09471_face1.jpg	F0897	MID4	F0897/MID4
4	../input/train/F0897/MID4/P09464_face3.jpg	F0897	MID4	F0897/MID4

- From below we can see that there are 470 families in total which comprises of 2316 total uniqueid with kins.

In [0]:
```
#[.nunique() gives the total unique values].
print("Total number of members in the dataset: {0}".format(train_images[
"uniqueId"].nunique()))
print("Total number of families in the dataset: {0}".format(train_images
["familyId"].nunique()))
```

```
Total number of members in the dataset: 2316
Total number of families in the dataset: 470
```

- from below we can see that

1. FamilyID="F0601" has maximum number of images i.e 776
2. person with uniqueID=F0601/MID6 has maximum number of images

In [0]:
```
#This section of code counts the maximum value.
family_with_most_pic = train_images["familyId"].value_counts()
kin_with_most_pic = train_images["uniqueId"].value_counts()
print("Family with maximum number of images: {0}, Image Count: {1}".form
at(family_with_most_pic.index[0], family_with_most_pic[0]))
print("Member with maximum number of images: {0}, Image Count: {1}".form
at(kin_with_most_pic.index[0], kin_with_most_pic[0]))
```

```
Family with maximum number of images: F0601, Image Count: 776
Member with maximum number of images: F0601/MID6, Image Count: 95
```

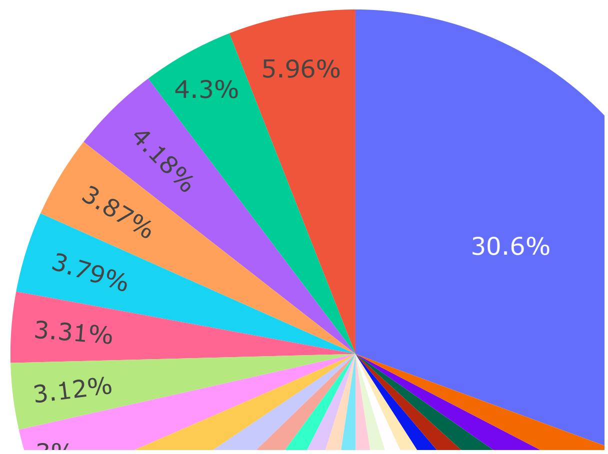
- out of various family we are calculating here top family with most pics

> REsult we found out that out of top families the highest pic that contains covers 30%, which can create bias

*[https://plot.ly/python/ipython-notebook-tutorial/ (https://plot.ly/python/ipython-notebook-tutorial/)](https://plot.ly/python/ipython-notebook-tutorial/)

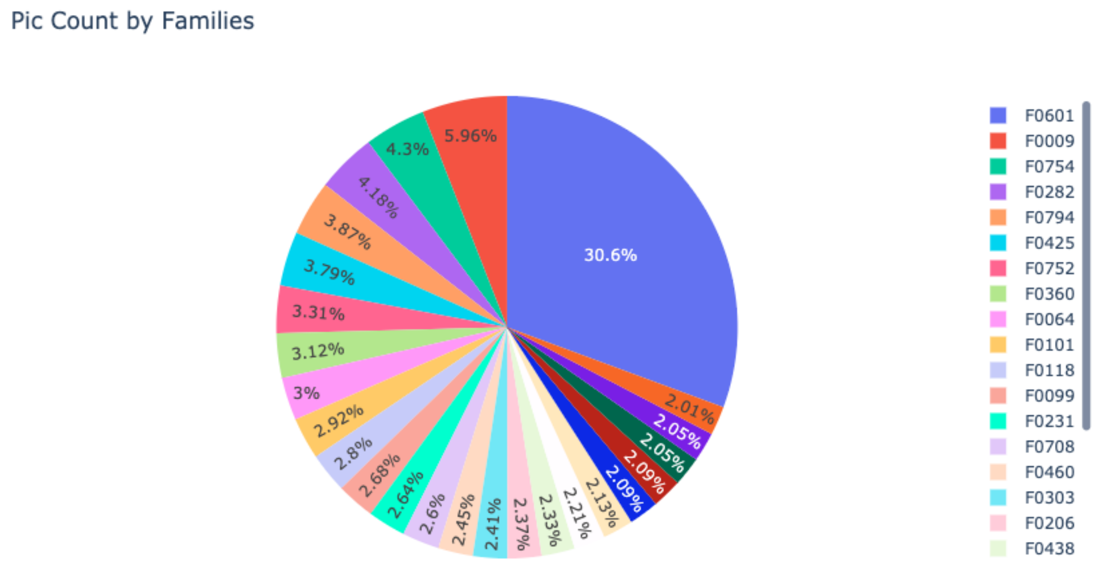
In [0]:
```
#taking 25 families from list
family_series = family_with_most_pic[:25]
labels = (np.array(family_series.index))
sizes = (np.array((family_series / family_with_most_pic.sum()) * 100))#c
alculating the percentage size

#calculating pipot from plotly
trace = go.Pie(labels=labels, values=sizes)
layout = go.Layout(title='Pic Count by Families')
data = [trace]
fig = go.Figure(data=data, layout=layout)
py.iplot(fig, filename='Families')
```

Pic Count by Families



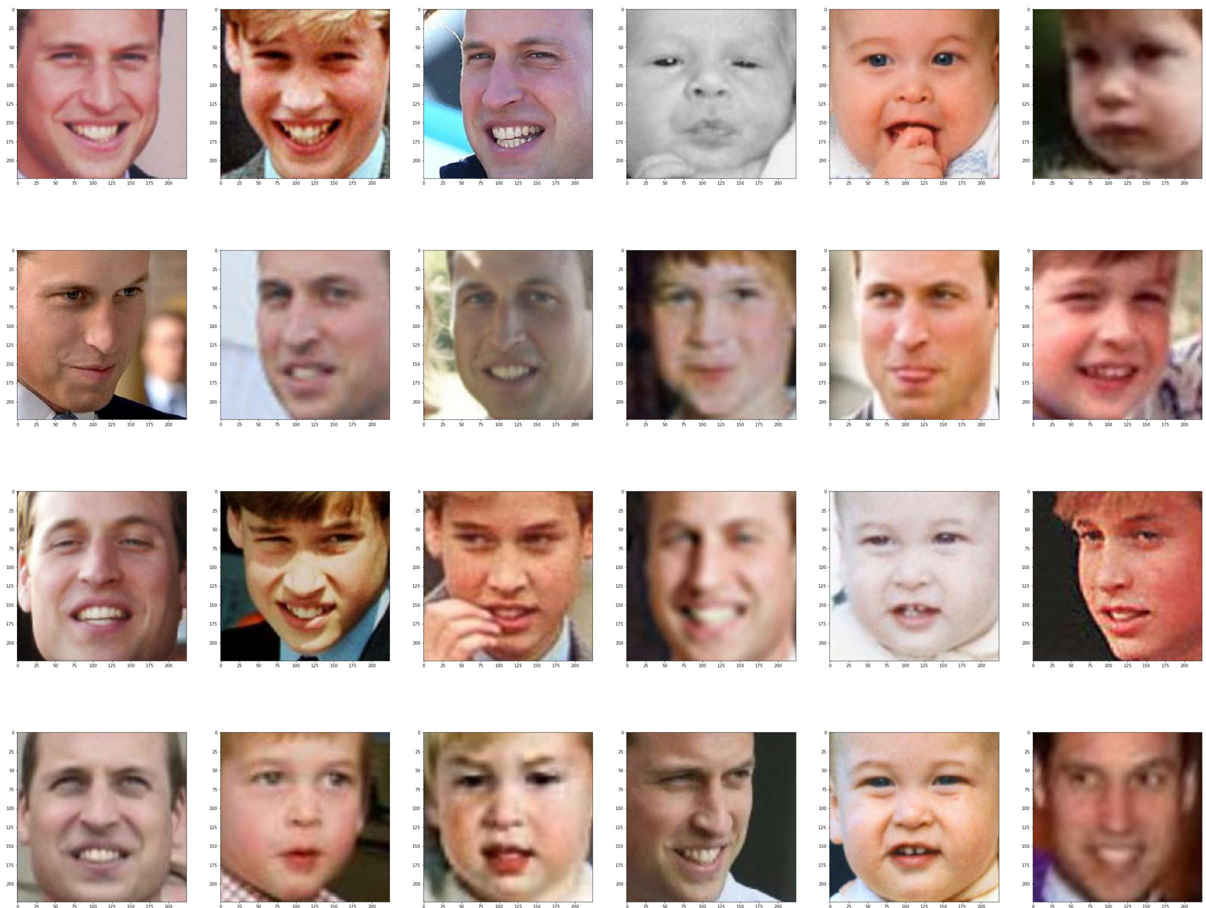
In [0]:
```
#https://stackoverflow.com/questions/10628262/inserting-image-into-ipython-notebook-markdown
from IPython.display import Image
Image(filename="/content/Screen Shot 2019-08-11 at 2.36.52 PM.png")
```

Out[0]:



Below we can see the person with most pics.*

In [0]:
```
#here we are matching pics with most unique id
most_pic_members = train_images[train_images["uniqueId"] == kin_with_mos
t_pic.index[0]].files.values
fig, ax = plt.subplots(4, 6, figsize=(50, 40))
row = 0
col = 0
for index in range(len(most_pic_members[:24])):
    with open(most_pic_members[index], 'rb') as f:
        #https://pillow.readthedocs.io/en/4.0.x/reference/open_files.htm
l
        #this block of code opens the image and shows it
        img = Image.open(f)
        ax[row][col].imshow(img)
        #for changing the rows and columns of image
        if(col < 5):
            col = col + 1
        else:
            col = 0
            row = row + 1
fig.show()
```



Checking families with maximum number of members and with least number of members. and we found out that maxm number of members in a family is 41. and minimum is 1.

In [0]:
```
#below codes arranges families with their unique id
family_with_most_members = train_images.groupby("familyId")["kinId"].nun
ique().sort_values(ascending=False)
print("Family with maximum number of members: {0}, Member Count: {1}".fo
rmat(family_with_most_members.index[0], family_with_most_members[0]))
print("Family with least number of members: {0}, Member Count: {1}".form
at(
    family_with_most_members.index[len(family_with_most_members)-1],
    family_with_most_members[len(family_with_most_members)-1]))
```

```
Family with maximum number of members: F0601, Member Count: 41
Family with least number of members: F0275, Member Count: 1
```

IN this section we are visualizing family with most members.

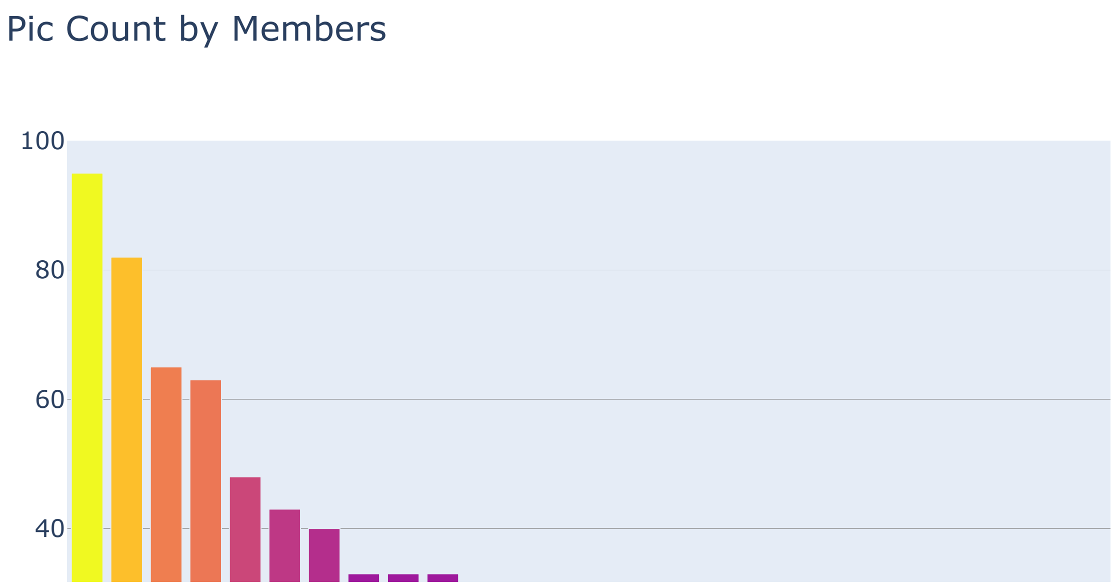
In [0]:
```
#loading family with maximum members.
large_family_df = train_images[train_images["familyId"]  == family_with_
most_members.index[0]]
large_family_df.head()
```

Out[0]:

	files	familyId	kinId	uniqueId
1199	../input/train/F0601/MID2/P11888_face1.jpg	F0601	MID2	F0601/MID2
1200	../input/train/F0601/MID2/P06281_face1.jpg	F0601	MID2	F0601/MID2
1201	../input/train/F0601/MID2/P06273_face1.jpg	F0601	MID2	F0601/MID2
1202	../input/train/F0601/MID2/P12068_face5.jpg	F0601	MID2	F0601/MID2
1203	../input/train/F0601/MID2/P11906_face4.jpg	F0601	MID2	F0601/MID2

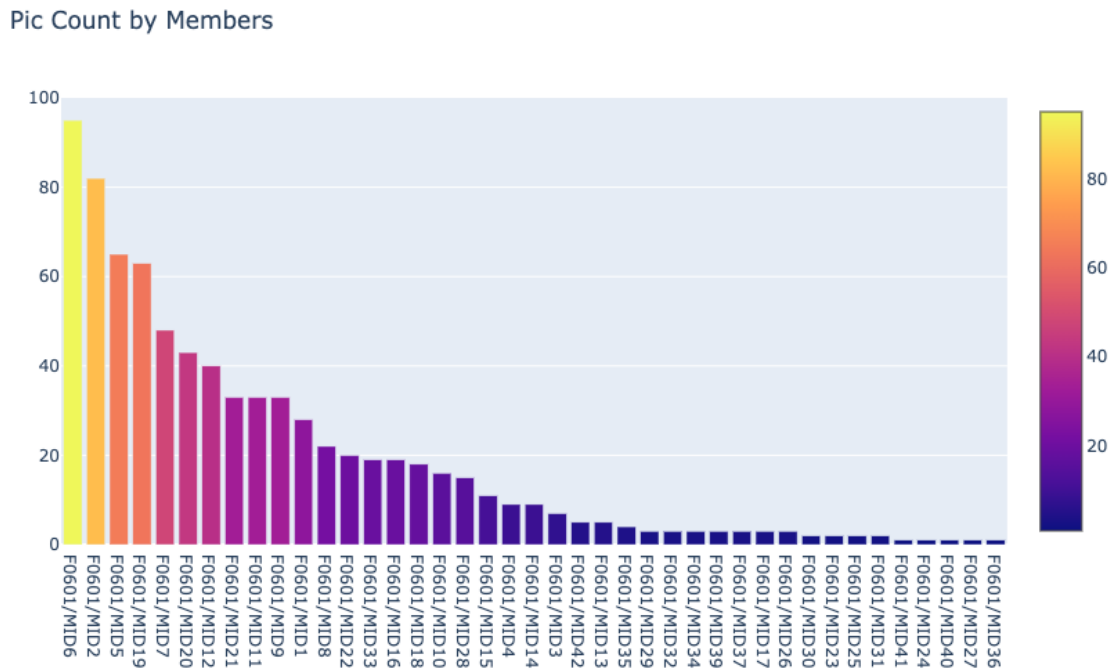
Pic Count of every member of largest family

In [0]:
```
#visualizing the number count by barplot
def render_bar_chart(data_df, column_name, title, filename):
    series = data_df[column_name].value_counts()
    count = series.shape[0]
    #https://plot.ly/python/ipython-notebook-tutorial/
    trace = go.Bar(x = series.index, y=series.values, marker=dict(
        color=series.values,
        showscale=True
    ))
    layout = go.Layout(title=title)
    data = [trace]
    fig = go.Figure(data=data, layout=layout)
    py.iplot(fig, filename=filename)

render_bar_chart(large_family_df, 'uniqueId', 'Pic Count by Members', 'm
embers')
```



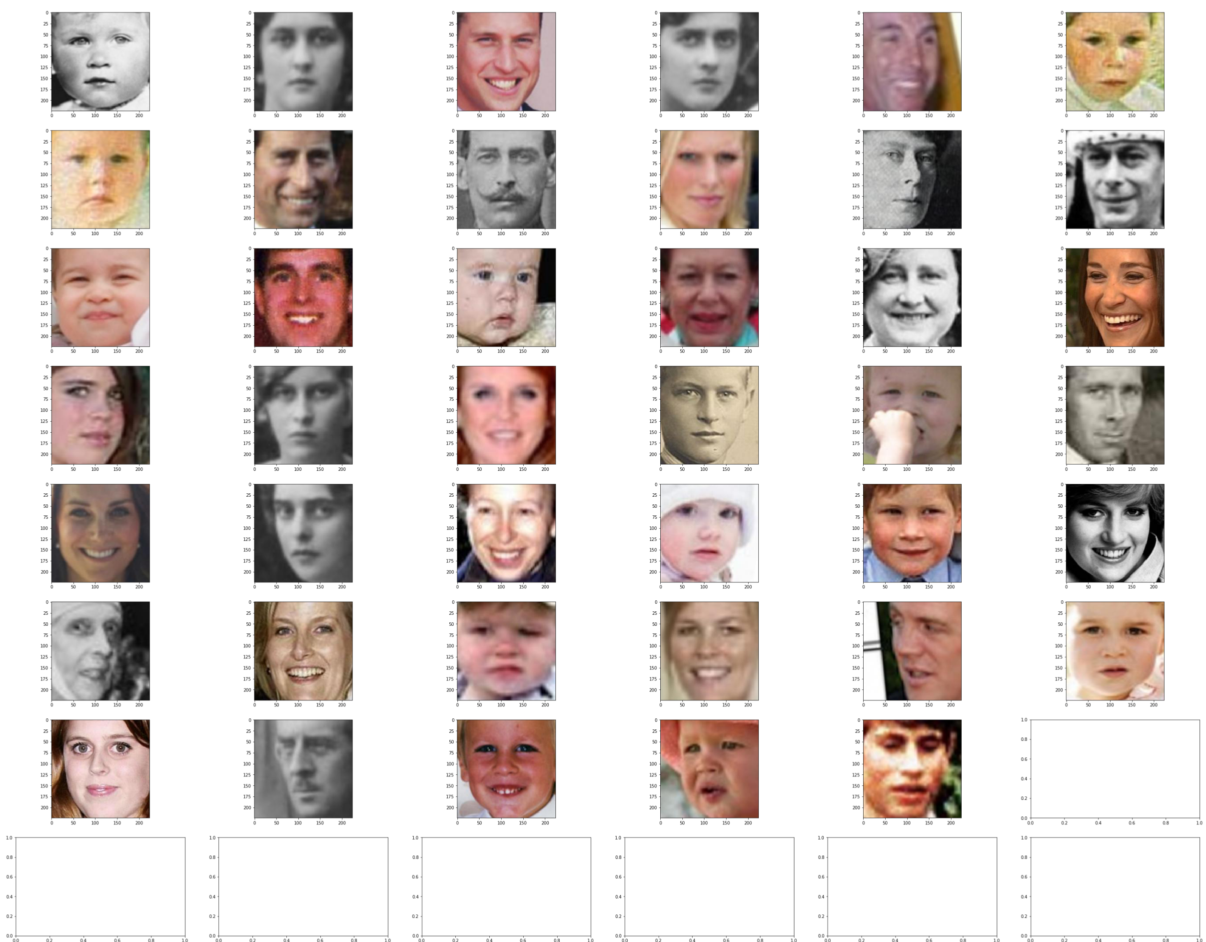
In [0]:
```python
#https://stackoverflow.com/questions/10628262/inserting-image-into-ipython-notebook-markdown
from IPython.display import Image
Image(filename="/content/Screen Shot 2019-08-11 at 2.06.19 PM.png")
```

Out[0]:



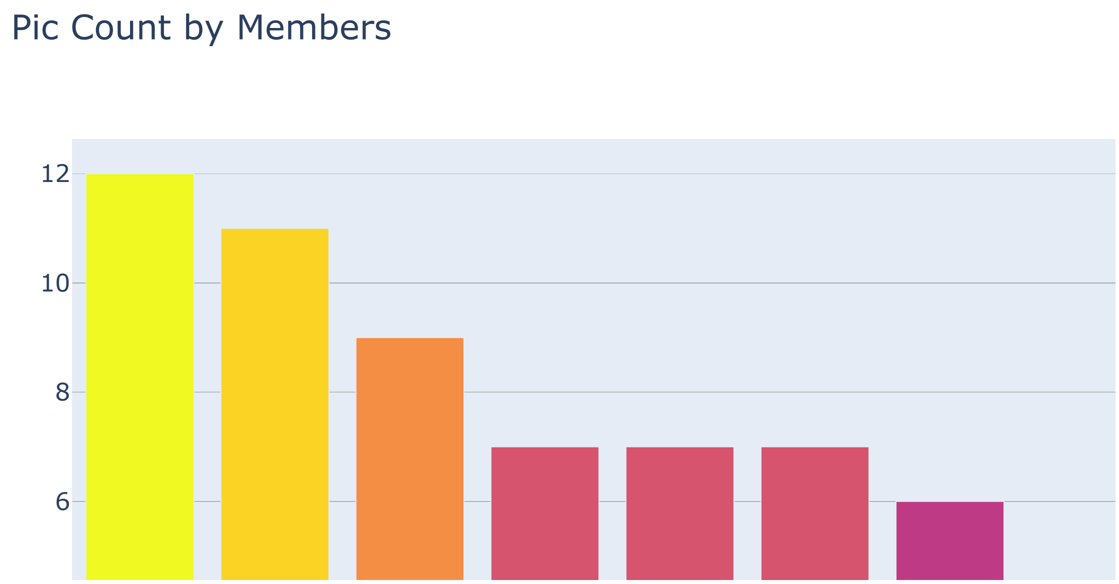
In [0]:

```
#image of largest family members
def render_images(large_family_df):
    large_family_pics = [large_family_df.loc[large_family_df.loc[large_family_df["uniqueId"] == aKin].index[0]]["files"] for aKin in large_family_df["uniqueId"].unique()]
    nrows = round(len(large_family_pics) / 6) + 1


    fig, ax = plt.subplots(nrows, 6, figsize=(50, 40))
    row = 0
    col = 0
    for index in range(len(large_family_pics)):
        with open(large_family_pics[index], 'rb') as f:
            img = Image.open(f)
            ax[row][col].imshow(img)

            if(col < 5):
                col = col + 1
            else:
                col = 0
                row = row + 1
    fig.show()
render_images(large_family_df)
```



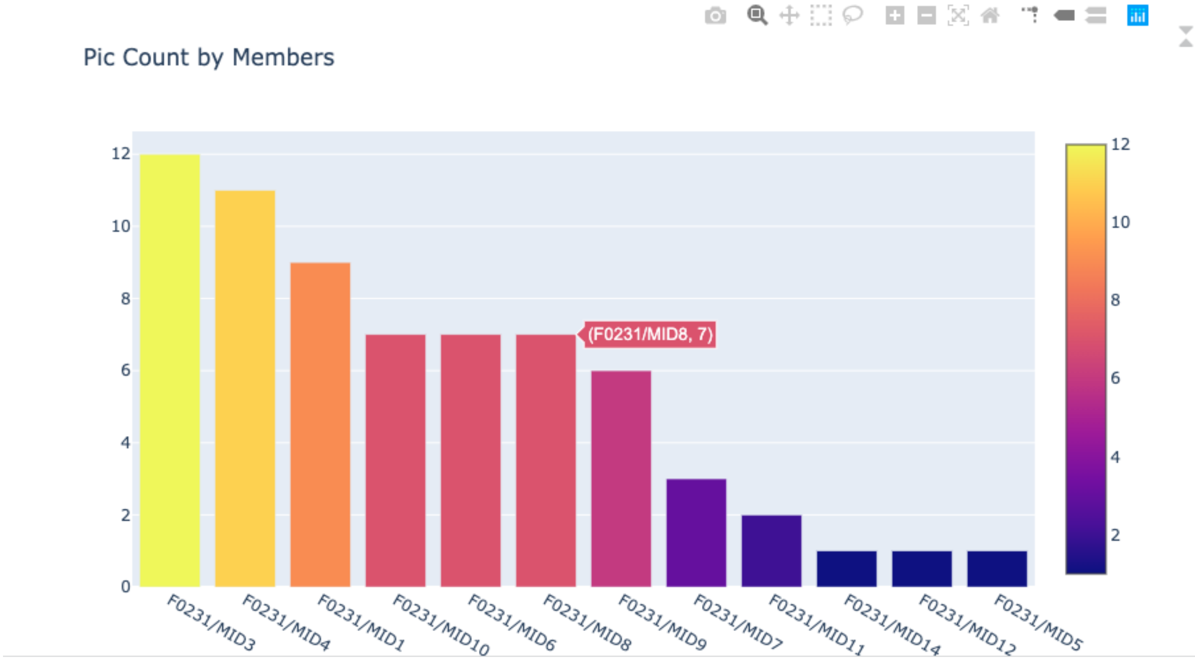
PIC count of every member of 5th last largest family

In [0]:
```
large_family_df = train_images[train_images["familyId"]  == family_with_most_members.index[4]]
render_bar_chart(large_family_df, 'uniqueId', 'Pic Count by Members', 'members')
```

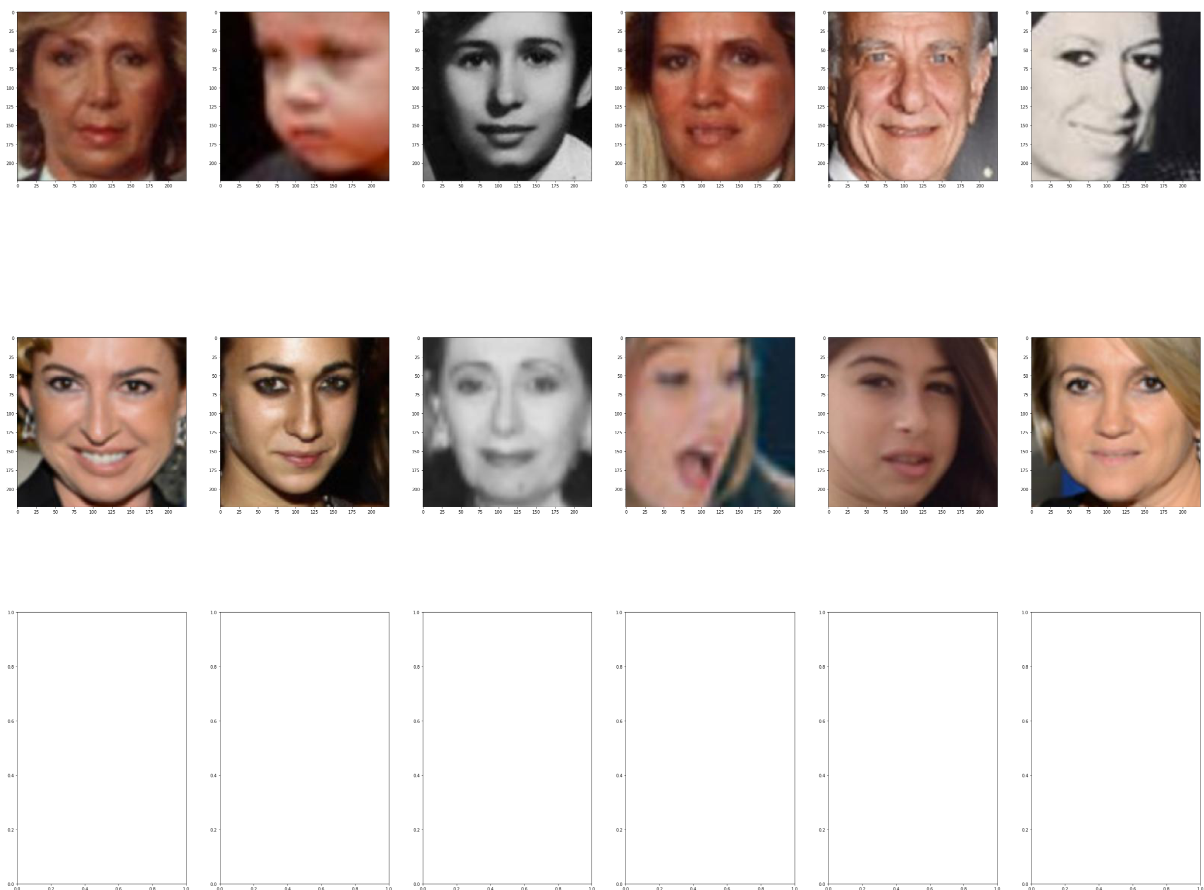


In [0]:
```
#https://stackoverflow.com/questions/10628262/inserting-image-into-ipython-notebook-markdown
from IPython.display import Image
Image(filename='/content/Screen Shot 2019-08-11 at 2.05.57 PM.png')
```

Out[0]:



In [0]:
```
render_images(large_family_df)
```



NOW MODELLING and TRAINING THE MODEL

In [0]:
```
train_file_path = "../input/train_relationships.csv"
train_folders_path = "../input/train/"
val_famillies = "F09"
```

splitting into train and validation image.

In [0]:
```
#train image takes all the files not in validation data.
train_images = [x for x in files if val_famillies not in x]
#validation image takes all the files in validation image.
val_images = [x for x in files if val_famillies in x]
```

there are 11232 train and 1147 val images in list.

In [0]:
```
len(train_images), len(val_images)
```

Out[0]:
```
(11232, 1147)
```

In [0]:
```
train_images[0:2]
```

Out[0]:
```
['../input/train/F0897/MID2/P09464_face1.jpg',
 '../input/train/F0897/MID5/P09469_face1.jpg']
```

makes dictionary with family unique id as keys and image of person as values

In [0]:
```
#making a dict from default list.
train_person_to_images_map = defaultdict(list)
ppl = [x.split("/")[-3] + "/" + x.split("/")[-2] for x in files]

#below codes makes dictionary with family unique id as keys and image of
person as values
for x in train_images:
    train_person_to_images_map[x.split("/")[-3] + "/" + x.split("/")[-2
]].append(x)

#similarly for validation also.
val_person_to_images_map = defaultdict(list)

for x in val_images:
    val_person_to_images_map[x.split("/")[-3] + "/" + x.split("/")[-2]].
append(x)
```

In [0]:
```
ppl[0:2]
```

Out[0]: ['F0897/MID2', 'F0897/MID5']

In [0]:
```
train_person_to_images_map["F0897/MID2"]
```

Out[0]: ['../input/train/F0897/MID2/P09464_face1.jpg']

1. train_df : making a list p1.values and p2.values and zipping it.

In [0]:
```
train_df = list(zip(train_df.p1.values, train_df.p2.values))
```

total data = 3362 bold text

In [0]:
```
train_df = [x for x in train_df if x[0] in ppl and x[1] in ppl]
len(train_df)
```

Out[0]: 3362

total train data=3066 and validation = 3362-3066=296

In [0]:
```
train = [x for x in train_df if val_famillies not in x[0]]
val = [x for x in train_df if val_famillies in x[0]]
len(train)
```

Out[0]: 3066

loading all the images from provided path and resizing it to size(197, 197) and then changing it to array
https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/load_img\
(https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/load_img\)
https://www.programcreek.com/python/example/89223/keras.preprocessing.image.load_img\
(https://www.programcreek.com/python/example/89223/keras.preprocessing.image.load_img\) and then finally passing the arrays to preprocess_input(it changes arrays to vgg format)\
https://stackoverflow.com/questions/47555829/preprocess-input-method-in-keras/47556342
(https://stackoverflow.com/questions/47555829/preprocess-input-method-in-keras/47556342)

In [0]:
```
def read_img(path):
    img = image.load_img(path, target_size=(197, 197))
    img = np.array(img).astype(np.float)
    return preprocess_input(img, version=2)
```

Defining a generator

since we have less dataset. [https://www.kaggle.com/shivamsarawagi/wildimagedetection-0-875 (https://www.kaggle.com/shivamsarawagi/wildimagedetection-0-875)](https://www.kaggle.com/shivamsarawagi/wildimagedetection-0-875)

In [0]:

```
def gen(list_tuples, person_to_images_map, batch_size=16):
    #taking the unique id
    ppl = list(person_to_images_map.keys())
    while True:
        batch_tuples = sample(list_tuples, batch_size // 2) #https://ww
w.geeksforgeeks.org/python-random-sample-function/
        labels = [1] * len(batch_tuples)
        while len(batch_tuples) < batch_size:
            #https://www.geeksforgeeks.org/python-numbers-choice-functio
n/
            p1 = choice(ppl) # returns random item.from choosen batch
            p2 = choice(ppl)

            if p1 != p2 and (p1, p2) not in list_tuples and (p2, p1) not
in list_tuples:
                batch_tuples.append((p1, p2))
                labels.append(0)

        for x in batch_tuples:
            if not len(person_to_images_map[x[0]]):
                print(x[0])

        #giving two inputs and one labels
        X1 = [choice(person_to_images_map[x[0]]) for x in batch_tuples]
        X1 = np.array([read_img(x) for x in X1])

        X2 = [choice(person_to_images_map[x[1]]) for x in batch_tuples]
        X2 = np.array([read_img(x) for x in X2])

        yield [X1, X2], labels
```

In [0]:
```
#defining the model.
def baseline_model():
    #taking input of shape(197, 197, 3)
    input_1 = Input(shape=(197, 197, 3))
    input_2 = Input(shape=(197, 197, 3))
    #input_3 = Input(shape=(197, 197, 3))

    #loading the vggface model with resnet50
    base_model = VGGFace(model='resnet50', include_top=False)

    #training the last 4 layer
    for x in base_model.layers[:-4]:
        x.trainable = True

    #passing input to basemodel
    x1 = base_model(input_1)
    x2 = base_model(input_2)

    #concatinating with last trainable model
    x1 = Concatenate(axis=-1)([GlobalMaxPool2D()(x1), GlobalAvgPool2D()(
x1)])
    x2 = Concatenate(axis=-1)([GlobalMaxPool2D()(x2), GlobalAvgPool2D()(
x2)])

    #subtracting input 1 and 2 and multiplying x3 to itself
    x3 = Subtract()([x1, x2])
    x3 = Multiply()([x3, x3])

    #multiplying and subtracting.
    x1_ = Multiply()([x1, x1])
    x2_ = Multiply()([x2, x2])
    x4 = Subtract()([x1_, x2_])

    #x5 = Multiply()([x4, x4])
    #x6 = Multiply()([x3, x4])

    x = Concatenate(axis=-1)([x3, x4])

    #making dense hidden layers
    x = Dense(100, activation="relu")(x)
    #x = BatchNormalization()(x)
    x = Dropout(0.01)(x)

    x = Dense(64, activation="relu")(x)
    #x = BatchNormalization()(x)
    x = Dropout(0.02)(x)

    out = Dense(1, activation="sigmoid")(x)

    model = Model(inputs=[input_1, input_2], outputs=[out])

    model.compile(loss="binary_crossentropy", metrics=['acc'], optimizer
=Adam(0.00001))

    model.summary()
```

```
    return model
```

In [0]:
```
#https://www.kaggle.com/shivamsarawagi/wildimagedetection-0-875
file_path = "vgg_face.h5"

checkpoint = ModelCheckpoint(file_path, monitor='val_acc', verbose=1, sa
ve_best_only=True, mode='max')

reduce_on_plateau = ReduceLROnPlateau(monitor="val_acc", mode="max", fac
tor=0.1, patience=20, verbose=1)

callbacks_list = [checkpoint, reduce_on_plateau]

model = baseline_model()
```

```
________________________________________________________________________________
__________________________
Layer (type)                    Output Shape         Param #     Connec
ted to
================================================================================
==========================
input_5 (InputLayer)            (None, 197, 197, 3)  0

________________________________________________________________________________
__________________________
input_6 (InputLayer)            (None, 197, 197, 3)  0

________________________________________________________________________________
__________________________
vggface_resnet50 (Model)        multiple             23561152    input_
5[0][0]
                                                                 input_
6[0][0]
________________________________________________________________________________
__________________________
global_max_pooling2d_3 (GlobalM (None, 2048)         0           vggfac
e_resnet50[1][0]
________________________________________________________________________________
__________________________
global_average_pooling2d_3 (Glo (None, 2048)         0           vggfac
e_resnet50[1][0]
________________________________________________________________________________
__________________________
global_max_pooling2d_4 (GlobalM (None, 2048)         0           vggfac
e_resnet50[2][0]
________________________________________________________________________________
__________________________
global_average_pooling2d_4 (Glo (None, 2048)         0           vggfac
e_resnet50[2][0]
________________________________________________________________________________
__________________________
concatenate_4 (Concatenate)     (None, 4096)         0           global
_max_pooling2d_3[0][0]
                                                                 global
_average_pooling2d_3[0][0]
________________________________________________________________________________
__________________________
concatenate_5 (Concatenate)     (None, 4096)         0           global
_max_pooling2d_4[0][0]
                                                                 global
_average_pooling2d_4[0][0]
________________________________________________________________________________
__________________________
subtract_3 (Subtract)           (None, 4096)         0           concat
enate_4[0][0]
                                                                 concat
enate_5[0][0]
________________________________________________________________________________
__________________________
multiply_5 (Multiply)           (None, 4096)         0           concat
enate_4[0][0]
                                                                 concat
enate_4[0][0]
________________________________________________________________________________
```

```
________________________________________
multiply_6 (Multiply)           (None, 4096)         0            concat
enate_5[0][0]
                                                                  concat
enate_5[0][0]

______________________________________________________________________________
________________________________________
multiply_4 (Multiply)           (None, 4096)         0            subtra
ct_3[0][0]
                                                                  subtra
ct_3[0][0]

______________________________________________________________________________
________________________________________
subtract_4 (Subtract)           (None, 4096)         0            multip
ly_5[0][0]
                                                                  multip
ly_6[0][0]

______________________________________________________________________________
________________________________________
concatenate_6 (Concatenate)     (None, 8192)         0            multip
ly_4[0][0]
                                                                  subtra
ct_4[0][0]

______________________________________________________________________________
________________________________________
dense_4 (Dense)                 (None, 100)          819300       concat
enate_6[0][0]

______________________________________________________________________________
________________________________________
dropout_3 (Dropout)             (None, 100)          0            dense_
4[0][0]

______________________________________________________________________________
________________________________________
dense_5 (Dense)                 (None, 64)           6464         dropou
t_3[0][0]

______________________________________________________________________________
________________________________________
dropout_4 (Dropout)             (None, 64)           0            dense_
5[0][0]

______________________________________________________________________________
________________________________________
dense_6 (Dense)                 (None, 1)            65           dropou
t_4[0][0]
==============================================================================
==========================
Total params: 24,386,981
Trainable params: 24,333,861
Non-trainable params: 53,120

______________________________________________________________________________
________________________________________
```

start training the model.

In [0]:
```
model.fit_generator(gen(train, train_person_to_images_map, batch_size=16
), use_multiprocessing=True,
                    validation_data=gen(val, val_person_to_images_map, b
atch_size=16), epochs=70, verbose=1,
                    workers = 4, callbacks=callbacks_list, steps_per_epo
ch=200, validation_steps=100)
```

```
Epoch 1/70
200/200 [==============================] - 92s 459ms/step - loss: 1.988
7 - acc: 0.5684 - val_loss: 1.8649 - val_acc: 0.5962

Epoch 00001: val_acc improved from -inf to 0.59625, saving model to vgg
_face.h5
Epoch 2/70
200/200 [==============================] - 62s 308ms/step - loss: 1.008
4 - acc: 0.6016 - val_loss: 1.1606 - val_acc: 0.6019

Epoch 00002: val_acc improved from 0.59625 to 0.60188, saving model to
vgg_face.h5
Epoch 3/70
200/200 [==============================] - 61s 307ms/step - loss: 0.755
7 - acc: 0.6328 - val_loss: 0.9491 - val_acc: 0.5975

Epoch 00003: val_acc did not improve from 0.60188
Epoch 4/70
200/200 [==============================] - 62s 309ms/step - loss: 0.652
2 - acc: 0.6556 - val_loss: 0.8418 - val_acc: 0.6269

Epoch 00004: val_acc improved from 0.60188 to 0.62687, saving model to
vgg_face.h5
Epoch 5/70
200/200 [==============================] - 62s 308ms/step - loss: 0.626
6 - acc: 0.6628 - val_loss: 0.8094 - val_acc: 0.6525

Epoch 00005: val_acc improved from 0.62687 to 0.65250, saving model to
vgg_face.h5
Epoch 6/70
200/200 [==============================] - 62s 309ms/step - loss: 0.586
7 - acc: 0.6878 - val_loss: 0.6710 - val_acc: 0.6994

Epoch 00006: val_acc improved from 0.65250 to 0.69937, saving model to
vgg_face.h5
Epoch 7/70
200/200 [==============================] - 61s 305ms/step - loss: 0.566
0 - acc: 0.7044 - val_loss: 0.7606 - val_acc: 0.6794

Epoch 00007: val_acc did not improve from 0.69937
Epoch 8/70
200/200 [==============================] - 60s 300ms/step - loss: 0.548
3 - acc: 0.7169 - val_loss: 0.6628 - val_acc: 0.6831

Epoch 00008: val_acc did not improve from 0.69937
Epoch 9/70
200/200 [==============================] - 61s 305ms/step - loss: 0.530
1 - acc: 0.7331 - val_loss: 0.6395 - val_acc: 0.6950

Epoch 00009: val_acc did not improve from 0.69937
Epoch 10/70
200/200 [==============================] - 59s 297ms/step - loss: 0.526
8 - acc: 0.7322 - val_loss: 0.6337 - val_acc: 0.7319

Epoch 00010: val_acc improved from 0.69937 to 0.73188, saving model to
vgg_face.h5
Epoch 11/70
```

```
200/200 [==============================] - 60s 300ms/step - loss: 0.488
6 - acc: 0.7637 - val_loss: 0.6207 - val_acc: 0.7119

Epoch 00011: val_acc did not improve from 0.73188
Epoch 12/70
200/200 [==============================] - 60s 298ms/step - loss: 0.492
3 - acc: 0.7550 - val_loss: 0.6059 - val_acc: 0.7350

Epoch 00012: val_acc improved from 0.73188 to 0.73500, saving model to
vgg_face.h5
Epoch 13/70
200/200 [==============================] - 59s 295ms/step - loss: 0.496
6 - acc: 0.7566 - val_loss: 0.5875 - val_acc: 0.7438

Epoch 00013: val_acc improved from 0.73500 to 0.74375, saving model to
vgg_face.h5
Epoch 14/70
200/200 [==============================] - 59s 294ms/step - loss: 0.461
1 - acc: 0.7759 - val_loss: 0.5352 - val_acc: 0.7462

Epoch 00014: val_acc improved from 0.74375 to 0.74625, saving model to
vgg_face.h5
Epoch 15/70
200/200 [==============================] - 59s 295ms/step - loss: 0.453
5 - acc: 0.7847 - val_loss: 0.5517 - val_acc: 0.7369

Epoch 00015: val_acc did not improve from 0.74625
Epoch 16/70
200/200 [==============================] - 59s 297ms/step - loss: 0.440
2 - acc: 0.7934 - val_loss: 0.5144 - val_acc: 0.7362

Epoch 00016: val_acc did not improve from 0.74625
Epoch 17/70
200/200 [==============================] - 60s 298ms/step - loss: 0.447
8 - acc: 0.7837 - val_loss: 0.5309 - val_acc: 0.7662

Epoch 00017: val_acc improved from 0.74625 to 0.76625, saving model to
vgg_face.h5
Epoch 18/70
200/200 [==============================] - 59s 293ms/step - loss: 0.417
8 - acc: 0.8047 - val_loss: 0.5552 - val_acc: 0.7375

Epoch 00018: val_acc did not improve from 0.76625
Epoch 19/70
200/200 [==============================] - 61s 303ms/step - loss: 0.438
4 - acc: 0.7906 - val_loss: 0.5252 - val_acc: 0.7431

Epoch 00019: val_acc did not improve from 0.76625
Epoch 20/70
200/200 [==============================] - 60s 298ms/step - loss: 0.409
7 - acc: 0.8081 - val_loss: 0.5397 - val_acc: 0.7519

Epoch 00020: val_acc did not improve from 0.76625
Epoch 21/70
200/200 [==============================] - 60s 298ms/step - loss: 0.402
6 - acc: 0.8134 - val_loss: 0.5243 - val_acc: 0.7538
```

```
Epoch 00021: val_acc did not improve from 0.76625
Epoch 22/70
200/200 [==============================] - 59s 297ms/step - loss: 0.402
5 - acc: 0.8131 - val_loss: 0.4864 - val_acc: 0.7788

Epoch 00022: val_acc improved from 0.76625 to 0.77875, saving model to
vgg_face.h5
Epoch 23/70
200/200 [==============================] - 59s 294ms/step - loss: 0.422
9 - acc: 0.7991 - val_loss: 0.5254 - val_acc: 0.7581

Epoch 00023: val_acc did not improve from 0.77875
Epoch 24/70
200/200 [==============================] - 60s 300ms/step - loss: 0.422
1 - acc: 0.8016 - val_loss: 0.5646 - val_acc: 0.7362

Epoch 00024: val_acc did not improve from 0.77875
Epoch 25/70
200/200 [==============================] - 59s 297ms/step - loss: 0.386
8 - acc: 0.8175 - val_loss: 0.5333 - val_acc: 0.7512

Epoch 00025: val_acc did not improve from 0.77875
Epoch 26/70
200/200 [==============================] - 60s 299ms/step - loss: 0.416
0 - acc: 0.8137 - val_loss: 0.4908 - val_acc: 0.7700

Epoch 00026: val_acc did not improve from 0.77875
Epoch 27/70
200/200 [==============================] - 59s 294ms/step - loss: 0.374
6 - acc: 0.8316 - val_loss: 0.5182 - val_acc: 0.7619

Epoch 00027: val_acc did not improve from 0.77875
Epoch 28/70
200/200 [==============================] - 59s 295ms/step - loss: 0.405
9 - acc: 0.8116 - val_loss: 0.4872 - val_acc: 0.7744

Epoch 00028: val_acc did not improve from 0.77875
Epoch 29/70
200/200 [==============================] - 59s 295ms/step - loss: 0.376
1 - acc: 0.8378 - val_loss: 0.5198 - val_acc: 0.7612

Epoch 00029: val_acc did not improve from 0.77875
Epoch 30/70
200/200 [==============================] - 59s 293ms/step - loss: 0.358
3 - acc: 0.8378 - val_loss: 0.4856 - val_acc: 0.7762

Epoch 00030: val_acc did not improve from 0.77875
Epoch 31/70
200/200 [==============================] - 59s 296ms/step - loss: 0.357
5 - acc: 0.8387 - val_loss: 0.5205 - val_acc: 0.7606

Epoch 00031: val_acc did not improve from 0.77875
Epoch 32/70
200/200 [==============================] - 59s 296ms/step - loss: 0.366
9 - acc: 0.8325 - val_loss: 0.4893 - val_acc: 0.7744

Epoch 00032: val_acc did not improve from 0.77875
```

```
Epoch 33/70
200/200 [==============================] - 59s 296ms/step - loss: 0.351
4 - acc: 0.8466 - val_loss: 0.5486 - val_acc: 0.7512

Epoch 00033: val_acc did not improve from 0.77875
Epoch 34/70
200/200 [==============================] - 60s 301ms/step - loss: 0.366
6 - acc: 0.8347 - val_loss: 0.5623 - val_acc: 0.7488

Epoch 00034: val_acc did not improve from 0.77875
Epoch 35/70
200/200 [==============================] - 60s 301ms/step - loss: 0.343
9 - acc: 0.8481 - val_loss: 0.5271 - val_acc: 0.7562

Epoch 00035: val_acc did not improve from 0.77875
Epoch 36/70
200/200 [==============================] - 60s 299ms/step - loss: 0.350
6 - acc: 0.8459 - val_loss: 0.5105 - val_acc: 0.7512

Epoch 00036: val_acc did not improve from 0.77875
Epoch 37/70
200/200 [==============================] - 60s 299ms/step - loss: 0.341
1 - acc: 0.8562 - val_loss: 0.5512 - val_acc: 0.7481

Epoch 00037: val_acc did not improve from 0.77875
Epoch 38/70
200/200 [==============================] - 60s 298ms/step - loss: 0.330
1 - acc: 0.8500 - val_loss: 0.4921 - val_acc: 0.7688

Epoch 00038: val_acc did not improve from 0.77875
Epoch 39/70
200/200 [==============================] - 60s 301ms/step - loss: 0.318
0 - acc: 0.8638 - val_loss: 0.4766 - val_acc: 0.7950

Epoch 00039: val_acc improved from 0.77875 to 0.79500, saving model to
vgg_face.h5
Epoch 40/70
200/200 [==============================] - 59s 297ms/step - loss: 0.328
9 - acc: 0.8512 - val_loss: 0.5462 - val_acc: 0.7594

Epoch 00040: val_acc did not improve from 0.79500
Epoch 41/70
200/200 [==============================] - 60s 301ms/step - loss: 0.331
7 - acc: 0.8559 - val_loss: 0.5319 - val_acc: 0.7400

Epoch 00041: val_acc did not improve from 0.79500
Epoch 42/70
200/200 [==============================] - 60s 299ms/step - loss: 0.317
7 - acc: 0.8613 - val_loss: 0.5434 - val_acc: 0.7488

Epoch 00042: val_acc did not improve from 0.79500
Epoch 43/70
200/200 [==============================] - 59s 295ms/step - loss: 0.318
6 - acc: 0.8566 - val_loss: 0.5617 - val_acc: 0.7525

Epoch 00043: val_acc did not improve from 0.79500
Epoch 44/70
```

```
200/200 [==============================] - 60s 299ms/step - loss: 0.311
3 - acc: 0.8709 - val_loss: 0.5276 - val_acc: 0.7644

Epoch 00044: val_acc did not improve from 0.79500
Epoch 45/70
200/200 [==============================] - 59s 296ms/step - loss: 0.302
7 - acc: 0.8669 - val_loss: 0.5164 - val_acc: 0.7681

Epoch 00045: val_acc did not improve from 0.79500
Epoch 46/70
200/200 [==============================] - 59s 295ms/step - loss: 0.289
2 - acc: 0.8750 - val_loss: 0.5279 - val_acc: 0.7681

Epoch 00046: val_acc did not improve from 0.79500
Epoch 47/70
200/200 [==============================] - 59s 294ms/step - loss: 0.298
5 - acc: 0.8700 - val_loss: 0.5622 - val_acc: 0.7700

Epoch 00047: val_acc did not improve from 0.79500
Epoch 48/70
200/200 [==============================] - 59s 294ms/step - loss: 0.317
9 - acc: 0.8613 - val_loss: 0.5234 - val_acc: 0.7506

Epoch 00048: val_acc did not improve from 0.79500
Epoch 49/70
200/200 [==============================] - 59s 297ms/step - loss: 0.284
5 - acc: 0.8803 - val_loss: 0.5399 - val_acc: 0.7619

Epoch 00049: val_acc did not improve from 0.79500
Epoch 50/70
200/200 [==============================] - 58s 292ms/step - loss: 0.293
6 - acc: 0.8712 - val_loss: 0.5649 - val_acc: 0.7488

Epoch 00050: val_acc did not improve from 0.79500
Epoch 51/70
200/200 [==============================] - 58s 292ms/step - loss: 0.301
9 - acc: 0.8738 - val_loss: 0.5553 - val_acc: 0.7494

Epoch 00051: val_acc did not improve from 0.79500
Epoch 52/70
200/200 [==============================] - 59s 294ms/step - loss: 0.281
0 - acc: 0.8788 - val_loss: 0.5688 - val_acc: 0.7625

Epoch 00052: val_acc did not improve from 0.79500
Epoch 53/70
200/200 [==============================] - 58s 292ms/step - loss: 0.272
6 - acc: 0.8872 - val_loss: 0.5706 - val_acc: 0.7456

Epoch 00053: val_acc did not improve from 0.79500
Epoch 54/70
200/200 [==============================] - 59s 295ms/step - loss: 0.285
5 - acc: 0.8797 - val_loss: 0.6135 - val_acc: 0.7550

Epoch 00054: val_acc did not improve from 0.79500
Epoch 55/70
200/200 [==============================] - 58s 292ms/step - loss: 0.291
2 - acc: 0.8759 - val_loss: 0.5041 - val_acc: 0.7612
```

```
Epoch 00055: val_acc did not improve from 0.79500
Epoch 56/70
200/200 [==============================] - 58s 292ms/step - loss: 0.294
3 - acc: 0.8719 - val_loss: 0.5102 - val_acc: 0.7725

Epoch 00056: val_acc did not improve from 0.79500
Epoch 57/70
200/200 [==============================] - 59s 294ms/step - loss: 0.268
7 - acc: 0.8959 - val_loss: 0.5457 - val_acc: 0.7544

Epoch 00057: val_acc did not improve from 0.79500
Epoch 58/70
200/200 [==============================] - 59s 294ms/step - loss: 0.260
6 - acc: 0.8972 - val_loss: 0.5978 - val_acc: 0.7394

Epoch 00058: val_acc did not improve from 0.79500
Epoch 59/70
200/200 [==============================] - 59s 295ms/step - loss: 0.267
9 - acc: 0.8931 - val_loss: 0.5435 - val_acc: 0.7756

Epoch 00059: val_acc did not improve from 0.79500

Epoch 00059: ReduceLROnPlateau reducing learning rate to 9.999999747378
752e-07.
Epoch 60/70
200/200 [==============================] - 59s 293ms/step - loss: 0.244
2 - acc: 0.9022 - val_loss: 0.5806 - val_acc: 0.7612

Epoch 00060: val_acc did not improve from 0.79500
Epoch 61/70
200/200 [==============================] - 60s 298ms/step - loss: 0.254
4 - acc: 0.8972 - val_loss: 0.5630 - val_acc: 0.7562

Epoch 00061: val_acc did not improve from 0.79500
Epoch 62/70
200/200 [==============================] - 59s 294ms/step - loss: 0.260
0 - acc: 0.8919 - val_loss: 0.6057 - val_acc: 0.7488

Epoch 00062: val_acc did not improve from 0.79500
Epoch 63/70
200/200 [==============================] - 59s 295ms/step - loss: 0.250
0 - acc: 0.8972 - val_loss: 0.5730 - val_acc: 0.7512

Epoch 00063: val_acc did not improve from 0.79500
Epoch 64/70
200/200 [==============================] - 59s 296ms/step - loss: 0.237
2 - acc: 0.9012 - val_loss: 0.5349 - val_acc: 0.7744

Epoch 00064: val_acc did not improve from 0.79500
Epoch 65/70
200/200 [==============================] - 59s 296ms/step - loss: 0.236
9 - acc: 0.9006 - val_loss: 0.5515 - val_acc: 0.7656

Epoch 00065: val_acc did not improve from 0.79500
Epoch 66/70
200/200 [==============================] - 59s 295ms/step - loss: 0.236
```

```
1 - acc: 0.9062 - val_loss: 0.6074 - val_acc: 0.7475

Epoch 00066: val_acc did not improve from 0.79500
Epoch 67/70
200/200 [==============================] - 59s 293ms/step - loss: 0.242
4 - acc: 0.9006 - val_loss: 0.5600 - val_acc: 0.7731

Epoch 00067: val_acc did not improve from 0.79500
Epoch 68/70
200/200 [==============================] - 59s 295ms/step - loss: 0.231
6 - acc: 0.9034 - val_loss: 0.6211 - val_acc: 0.7619

Epoch 00068: val_acc did not improve from 0.79500
Epoch 69/70
200/200 [==============================] - 59s 294ms/step - loss: 0.229
4 - acc: 0.9119 - val_loss: 0.5813 - val_acc: 0.7588

Epoch 00069: val_acc did not improve from 0.79500
Epoch 70/70
200/200 [==============================] - 59s 294ms/step - loss: 0.224
1 - acc: 0.9113 - val_loss: 0.5972 - val_acc: 0.7612

Epoch 00070: val_acc did not improve from 0.79500
```

Out[0]:

```
<keras.callbacks.History at 0x7fe07e40eba8>
```

Stacking

credit https://www.kaggle.com/vaishvik25/blend-of-smiles\ (https://www.kaggle.com/vaishvik25/blend-of-smiles\) **Submission4 is taken from above link**

In [0]:

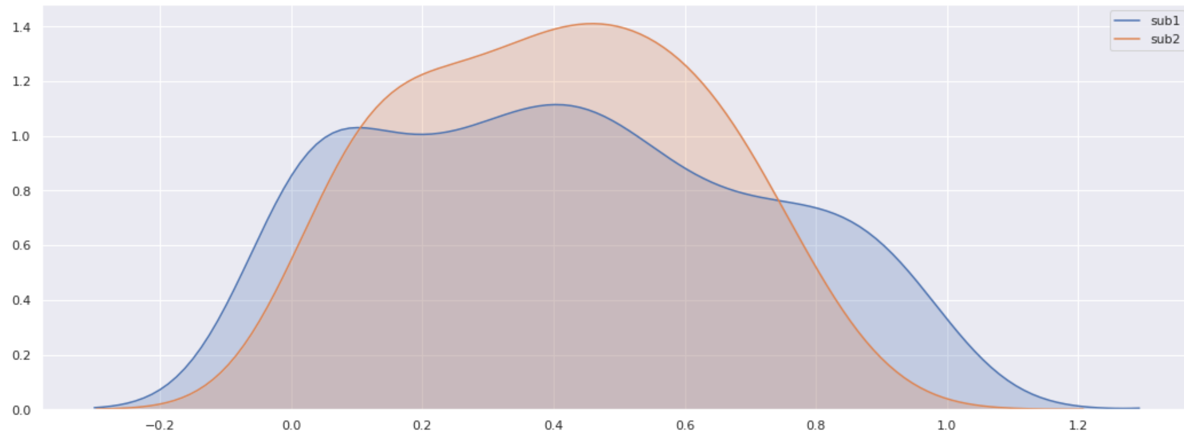
```
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from subprocess import check_output
```

In [0]:

```
sub1 = pd.read_csv('/content/stack_mean.csv')
sub2 = pd.read_csv('/content/submission4.csv')
```

In [0]:
```
sns.set(rc={'figure.figsize':(18,6.5)})
sns.kdeplot(sub1['is_related'],label="sub1",shade=True,bw=.1)
sns.kdeplot(sub2['is_related'], label="sub2",shade=True,bw=.1)
```

Out[0]: <matplotlib.axes._subplots.AxesSubplot at 0x7fcf8d14b9e8>



In [0]:
```
temp=pd.read_csv('stack_mean.csv')
```

In [0]:
```
temp['is_related'] = 0.5*sub1['is_related'] + 0.5*sub2['is_related']
temp.to_csv('submission5.csv', index=False )
```

In [0]:
```
#https://www.kaggle.com/vaishvik25/blend-of-smiles
```

Conclusion

On test data this model performed better than val_data and tweaking output a little bit and with lots of output, model performed much better\ training last 4 layers gives good results than 3 in val data. but when it comes to test 3 performs better\ stacking results in rank under 20.

Steps Done:

1. Importing necessary libraries.
2. Rcmallai vggnet
3. loading csv
4. Creating dataframe.
5. putting all images in file
6. Visualizing the person with most pic
7. Checking families with most member and least member.
8. Visualiging yop and 5th family.
9. seperating images into train and test.
10. Defining a generator.
11. defining a baseline model with vggnet.
12. start training model